1782-JDM
DeviceNet™-ModBus Gateway
User's Manual

Western Reserve Controls, Inc.

Although every effort has been made to insure the accuracy of this document, all information is subject to change without notice. WRC takes no liability for any errors in this document or for direct, indirect, incidental or consequential damage resulting from the use of this manual.

Document PUB 24.0
Rev 2.11
July 2011

Copyright © 1999-2011 WRC

Western Reserve Controls, Inc.
1485 Exeter Road
Akron OH 44306
330-733-6662 (Phone)
330-733-6663 (FAX)
sales@wrcakron.com (Email)
http://www.wrcakron.com (Web)

1782-JDM, JDM and WRC are trademarks of Western Reserve Controls.
DeviceNet is a trademark of ODVA, Inc.
ModBus is a trademark of Schneider Automation Inc.
All other trademarks are property of their respective companies.

TABLE OF CONTENTS

1 OVERVIEW 3

1.1 FEATURES 4
1.1 PRODUCT FAMILY SELECTION 4
1.2 DEVICENET SYSTEM ARCHITECTURE 4
1.3 JDM OPERATIONAL PHILOSOPHY 5
1.4 JDM BASIC OPERATION 5
1.4.1 Polled I/O 6
1.4.2 Explicit Messages 7
1.5 DEFAULT DEVICE CONFIGURATION 7
1.6 EDS 7

2 QUICK START 8

2.1 HOW TO CHANGE THE NODE ADDRESS 8
2.2 HOW TO CHANGE THE DEVICENET BAUD RATE 9
2.3 HOW TO INSTALL THE MODBUS DEVICE NETWORK 9
2.4 HOW TO READ MODBUS DEVICE INPUT DATA 10
2.5 HOW TO WRITE MODBUS DEVICE OUTPUT DATA 11

3 HARDWARE INSTALLATION AND CONFIGURATION 13

3.1 OVERVIEW 13
3.2 LED OPERATION 14
3.3 RS232/RS485 CONNECTOR 15
3.3.1 RS485 Network Termination (1782-JDC-2) 15
3.4 POWER REQUIREMENTS 15
3.4.1 Device power 15
3.5 DEVICENET NETWORK CONFIGURATION 16
3.5.1 DeviceNet Network Termination 17
3.5.2 DeviceNet Connection Wiring 17

4 SOFTWARE CONFIGURATION 19

5 DEVICENET I/O ASSEMBLY FORMATS 22

5.1 OUTPUT DATA, MODBUS COMMAND 22
5.2 INPUT DATA, MODBUS RESPONSE 23

6 GENERAL SPECIFICATIONS 24

7 MODBUS PROTOCOL 26

8 TROUBLESHOOTING 27

9 ACCESSORIES AND OTHER WRC PRODUCTS 29

10 FREQUENTLY ASKED QUESTIONS 30

LIST OF TABLES

TABLE 1-1 DATA MESSAGE OPTIONS 6
TABLE 2-1 ADDRESS SELECTION 8
TABLE 2-2 RS-232/485 CONNECTOR SIGNALS 9
TABLE 2-3 RS-422 CONNECTOR SIGNALS 9
TABLE 2-3 POLL RESPONSE STATUS BYTE CODES 11
TABLE 3-1 MODULE STATUS LED (LABELED MS) 14
TABLE 3-2 NETWORK STATUS LED (LABELED NS) 14
TABLE 3-3 RS-232/485 CONNECTOR SIGNALS 15
TABLE 3-4 RS-422 CONNECTOR SIGNALS 15
TABLE 3-5 NETWORK MAXIMUM LENGTHS 16
TABLE 3-6 DEVICENET WIRING TERMINATION 17
TABLE 3-7 DEVICENET CONDUCTOR SIZES 18
TABLE 4-1 CONFIGURATION PARAMETERS 20
TABLE 5-1 I/O ASSEMBLY OBJECT INSTANCE 2 DATA FORMAT – FRAGMENTED COMMAND 22
TABLE 5-2 I/O ASSEMBLY OBJECT INSTANCE 2 DATA FORMAT – SHORT COMMAND W/O FRAGMENTATION 23
TABLE 5-3 I/O ASSEMBLY OBJECT INSTANCE 1 DATA FORMAT – RESPONSE 23
TABLE 9-1 WRC REPLACEMENTS, SPARE PARTS AND OTHER PRODUCTS 30

LIST OF FIGURES

FIGURE 1-1 1782-JDM 3
FIGURE 3-1 1782-JDM OUTLINE 13
FIGURE 3-2 DEVICENET CABLE CONNECTOR 18
FIGURE 3-3 DEVICENET CABLE SPECIFICATIONS 18

1 Overview

The 1782-JDM is a family of DeviceNet-to-ModBus interfaces that allows you to easily connect and integrate a ModBus-capable device into a DeviceNet system.

The ModBus device's real-time data is read and written as I/O by the DeviceNet Master via Polled I/O. The devices set up parameters, if any, can be read and written with the Explicit Messaging technique. The 1782-JDM will operate with any compliant DeviceNet master software system or other device.

The JDM is defined as a Communications Adapter device on the DeviceNet system. It has a 3-pin plug connector for RS232/485 or a 5-pin connector for RS422 to cable to the ModBus interface port on the MODBUS DEVICE and a 5-pin pluggable DeviceNet connector for connections to the DeviceNet network. The device performs the DeviceNet baud rate selection automatically when it is powered up on a network. The JDM has one assigned DeviceNet address that is set by a 6-position DIP switch on the unit. Other parameters are software-configurable and are changed from their default values by third-party DeviceNet configuration tools. Each JDM has 2 standard green/red DeviceNet LED's for module status and network status and two green LED's to indicate RS232/485/422 transmit and receive activity.

All the ModBus device parameters are accessible for read or write over DeviceNet via the JDM, which allows the user the option of using DeviceNet tools and software to define or monitor these parameters in addition to other off-the-shelf or custom software utility using the ModBus port connection.

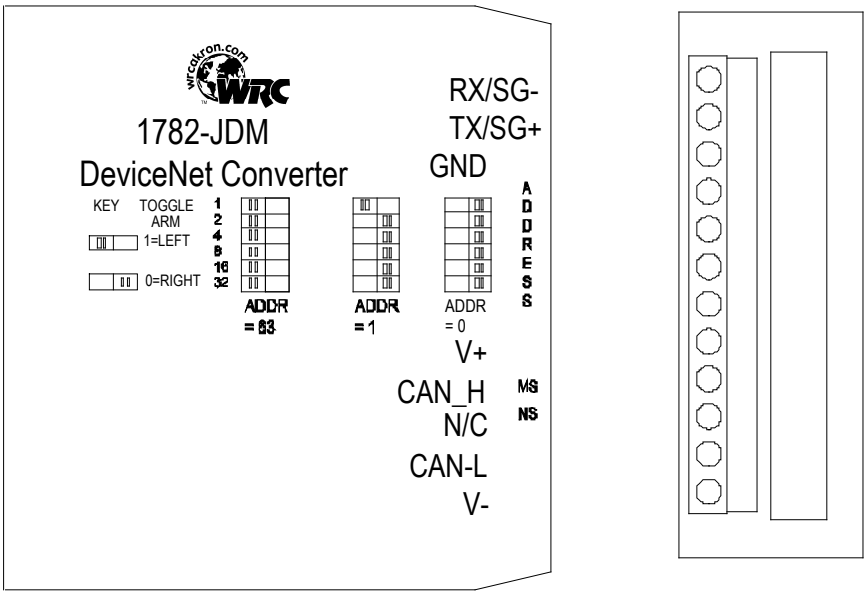

Figure 1-1 1782-JDM

1.1 Features

The JDM has the following features:

- Translates messages and data between DeviceNet and ModBus devices
- Operates independently of ModBus device
- Forwards the message received without interpretation
- Each JDM communicates with one or more ModBus devices
- DeviceNet Autobaud operation
- Polled I/O
- Explicit Messaging
- All ModBus registers on target device are accessible
- Upgrades to the ModBus device do not require updates to the JDM
- Software Configurable Parameters for device operation
- Address selection via DIP switches
- DIN rail mount
- Pluggable 5-pin DeviceNet connection
- Pluggable RS232/RS485 3-pin connection or RS422 5-pin connection
- Choice of isolated RS232, RS485 or RS422 serial interface
- Standard DeviceNet module and network status LED's
- Serial port transmit and receive LED's
- Powered from DeviceNet network

1.1 Product Family Selection

There are 3 members of the 1782-JDC family:

- o 1782-JDM-1 Isolated RS-232
- o 1782-JDM-2 Isolated RS-485
- o 1782-JDM-4 Isolated RS-422

1.2 DeviceNet System Architecture

A DeviceNet network is a distributed I/O system that may contain many different products from several different vendors. Products may be configured uniformly, as clusters or as distributed clusters. Up to 64 devices, including the master, may be attached to a single DeviceNet network. Any of these, except the master, may be a JDM. A typical system will include a master, such as a PLC or industrial PC, and multiple slave devices, including a JDM with connected ModBus devices.

1.3 JDM Operational Philosophy

A Modbus network system consists of a single MB Master and one or more MB Slave Devices. The JDM, as the proxy to the DeviceNet Master, becomes the ModBus master. All other devices on the Modbus link must be defined as Slaves.

- The JDM is a Slave on the DeviceNet bus

- The JDM communicates on DN with the DN Master using the Polled I/O technique for transmitting and receiving MB data.

- As just mentioned, the JDM acts as the Master on the Modbus line. It initiates all communications (commands) to the MB devices, which are generated by the DeviceNet Master.

- All Modbus devices connected to it must be set up as slaves, which only communicate in response to the MB commands issued by the DeviceNet Master (PC, PLC, controller, etc.) thru the JDM.

- Each MB device on the serial line must be set to a unique MB address, 0-255. The JDM, as MB link master, does not get a MB address assigned to it.

- The DN master software program must form a valid MB command string in a data array, pre-pend a Record Number and String Byte Length, and then send the string to the JDM. The string must contain:
    - slave MB address which is being addressed for this command string
    - MB command number
    - slave starting register for the command
    - and typically, depending upon the command number and command format, the number of registers referenced
    - and any data required (for outputs, for example)

In order to accomplish this, the user must understand how to generate an appropriate ModBus command. The format of this string generally follows the format of a MB command in RTU mode (hexadecimal) format. (See Table 5-1 in the Manual.)

- The Master then transmits this string to the JDM. The JDM will interrogate the Record Number, and, if the Record number is different from the previous record, will format the MB command string and transmit the new command out the serial port.

- The JDM does not interpret any data sent to it from the DN Master. It simply reformats the command string for either ASCII or RTU mode MB data transfers (as defined in Parameter 1 of the JDM) to send to the MB slaves. (For instance in RTU mode the JDM calculates and appends the 2-byte CRC to the end of the string received from the Master.)

- The JDM then listens for the MB response. When received, the JDM checks the serial port status (parity, overflow, etc.), checks and strips the CRC/LRC, reformats the data (per Table 5-2 in the Manual), pre-pends a new RX record number, a status byte and a Length byte, and sends the resultant string to the DN Master using a DeviceNet Poll Response.

- The DN Master's software then has the task of deciphering the received string and taking appropriate action.

1.4 JDM Basic Operation

The JDM operates as the DeviceNet front-end gateway between the DeviceNet system and the ModBus devices. It is a DeviceNet server device that can be assigned by the system implementer to one specific

master. The DeviceNet Master can receive and send data to and from the JDM via the methods described in this section. It interprets messages received from the DeviceNet connection, and if it is a ModBus message, reformats and sends it to the ModBus device. Likewise, it accepts and interprets responses from the ModBus device, which are reformatted and passed back to the DeviceNet system as required.

The JDM has one DeviceNet address. All Explicit DeviceNet messages to the JDM itself (to read / write its internal data) are sent to this address. The ModBus devices are communicated to using poll messages sent to the JDM and poll responses received from the JDM. See Section 2.4 and 2.5 for details.

The JDM Parameter Object allows you to define the specific operation of each JDM. These parameters include set-up for the serial comm link and the fault and idle actions and data. See Section 4 for details.

Each ModBus device connected to the JDM has a ModBus address between 1 and 247. This address is used by the host system and the JDM to identify which device on the ModBus network is being selected for each message.

Table 1-1 Data Message Options

I/O Type	Polled	Cyclic	Bit-Strobe	Change-of-State	Explicit Message
ModBus Device Commands	√				
ModBus Device Status	√				
JDM Parameters					√

The following sections describe how the data is accessed.

1.4.1 Polled I/O

The master can poll the run-time data used in the JDM. The communications is a two-part transaction: there is a Poll Command from the host to the JDM and the JDM responds with a Poll Response using the response of the addressed ModBus device.

How do I send a poll command to write data?

The poll command can set multiple registers of data per output command, depending upon the ModBus command used. First, define the ModBus command string desired to send to the ModBus device. Enter this data in binary format – the normal format for RTU mode. This format is also used if in ModBus ASCII mode – the JDM will translate from binary to ASCII. Do not include the leading colon (for ASCII mode) or the CRC or LRC. Determine the number of bytes to be written, with the data in binary format. Then set up your scanner to write the defined number bytes of data to the JDM **plus two (2)**.

Any ModBus command and register that is appropriate to the target ModBus device are valid. See the ModBus device's user's manual for more details on which commands are supported.

How do I send a poll command to read data?

The poll command can read multiple ModBus Device status registers' data per poll command, as supported by the device, up to a limit of 100 bytes of data on the DeviceNet message. First, set up your scanner to read the number of bytes of data expected from the JDM **plus three (3)**, depending upon the

command and number of register data desired. The command response from the ModBus device is returned from the JDM to the DeviceNet master in response to a poll command.

The DeviceNet response data includes a record number byte and a status byte, which contains a "waiting" bit. The DeviceNet response will be valid when both the record number matches the one sent in the Poll Command and the waiting bit (bit 7) is reset to zero (0).

1.4.2 Explicit Messages

As mentioned explicit messages are typically used to read and write configuration data. See the DeviceNet Specification for further information of the standard data available.

1.5 Default Device Configuration

The JDM DeviceNet address is read from the switches and is set to 63 at the factory. All other parameters are software settable. The default settings for the JDM are provided in Section 4.

1.6 EDS

An EDS (Electronic Data Sheet), which describes the various parameters of the JDM, is shipped with your device or is available on WRC's web site: http://www.wrcakron.com/

2 Quick Start

To quickly and easily install your JDM in your DeviceNet system, follow the instructions below. For more details, see Section 2.3.

To Install and Establish DeviceNet Communications

1. Connect the DeviceNet cable to the 5-pin plug (supplied) according to DeviceNet cable wiring specifications.
2. Make sure that the DeviceNet network is terminated properly. (See Section 3.5 below.)
3. The JDM Node Address (MacID) is set to 63 at the factory. Make sure no other device on the network is set to 63, or change the JDM address to one that is not currently used (see Section 2.1).
4. Use the autobaud feature of the JDM.
5. Make sure that there is power on the DeviceNet network and that it is plugged into a Master device.
6. Plug the DeviceNet cable into the JDM.
7. The JDM will undergo its initialization sequence, flashing both LEDs. After approximately 4 seconds, the Module Status LED (labeled "MS") will go on solid green and network LED will flash green.
8. The green Network Status LED (labeled "NS") will go on solid once the Master recognizes the unit on the link and allocates the connection.
9. The JDM is now operating on the network.

2.1 How to Change the Node Address

1. The address is set using the 6-position DIP switch block. They define the address as follows:

Table 2-1 Address Selection

Node Address	Switch Positions					
	1	2	3	4	5	6
	Switch Position Values					
	32	16	8	4	2	1
0	OFF	OFF	OFF	OFF	OFF	OFF
1	OFF	OFF	OFF	OFF	OFF	**ON**
2	OFF	OFF	OFF	OFF	**ON**	OFF
3	OFF	OFF	OFF	OFF	**ON**	**ON**
4	OFF	OFF	OFF	**ON**	OFF	OFF
5	OFF	OFF	OFF	**ON**	OFF	**ON**
…						
62	**ON**	**ON**	**ON**	**ON**	**ON**	OFF
63	**ON**	**ON**	**ON**	**ON**	**ON**	**ON**

OFF = OPEN = O
ON = CLOSED = 1

Note: Some parts may ship with switches that read OPEN/CLOSED and some with OFF/ON.

2. Switch setting changes do NOT take effect until the device is reset with either a device RESET command (to the class 1 Identity object) or a power cycle.

2.2 How to Change the DeviceNet Baud Rate

1. The baud rate defaults to autobaud. You can change the baud rate in software to a fixed baud rate using your configuration tool or an explicit message to Set Class 3 (DeviceNet Class), Instance 1, Attribute 2 to 1 (125k), 2 (250k), or 3 (500k), from the Master.

2. If you change the baud rate, the new baud rate will not become effective until the unit is power cycled or a Reset command is issued to the JDM from the Master.

2.3 How to Install the ModBus Device Network

1. The communication between the ModBus device(s) and the 1782-JDM is an RS-232 3-wire full-duplex link , an RS-485 network, using a 2-wire half-duplex differential signal between the devices, or an RS-422 5-wire full-duplex link.

2. Connect an appropriate cable to your ModBus device. See your ModBus device users manual for details.

3. Connect the other end of the cable to the JDM using the 3-point or 5-point terminal plug provided. See Table 2.2 and 2.3 below.

4. Turn on power to the ModBus device.

Table 2-2 RS-232/485 Connector Signals

Pin #	RS232 Signal	RS485 Signal	Designator
1	Receive	Signal -	RX/SIG-
2	Transmit	Signal +	TX/SIG+
3	Ground	Ground	GND

Table 2-3 RS-422 Connector Signals

Pin #	Designator	RS422 Signal
5	TX-	Transmit -
4	TX+	Transmit +
3	GND	Ground
2	RX-	Receive -
1	RX+	Receive +

2.4 How to Read ModBus Device Input Data

1. Set up the poll input data field size in your Master's scan table to be equal to or greater than the number of bytes required for the largest ModBus message received from your device(s). (Default size is 48 bytes).

2. Set up a Poll Connection (Allocate Connection Instance 2) to the 1782-JDM from the Master.

3. Configure the serial port to read input data by setting the appropriate values in the Parameter Data, described in Section 4.

4. Perform a poll function to the JDM using one of the ModBus read command strings. This will contain the "record number" in the 1st byte (which must be different than the last sent message), followed by a length byte and then the command string as shown in the ModBus RTU specification, WITHOUT the leading colon, the checksum, or the carriage return/linefeed. The JDM will translate this command to the proper ModBus format, compute the checksum, add the colon, carriage return/linefeed (ASCII mode), and send the message out the serial port. The JDM will then return a response with an updated record number (indicating that it has received and processed the previous command) and with the "waiting" bit set in the status byte (indicating that a response to that command has not yet been received from the ModBus device).

5. Once the device's response is received by the JDM, the JDM will check the checksum, and if ok will strip the leading colon, the checksum, and the carriage return/linefeed from the message (in ASCII mode). It will then produce a poll response with the "waiting" bit reset.

6. The response includes three extra bytes of data in addition to the ModBus data. The first byte of the response will be a "record counter" which is a counter generated by the JDM to mark the message with a sequential count value. Each time a new message is received at the JDM from the ModBus network, it will increment the internal record number counter.

7. The second byte of the record will be a status byte. This is the status of the ModBus transaction. It will be non-zero if an error occurred, or zero if OK. See for specific values.

8. The third byte is length, indicating the number of valid ModBus data bytes in the following data field.

9. The response Modbus data bytes will be returned in the mapped poll response data field. The data will be in binary hexadecimal format (ASCII 'A' = 41 hex, etc) and will represent the complete ModBus message, less checksum / carriage return/linefeed and the leading ":" (colon) (for ASCII mode). This data will also be formatted such that each data word contains the least significant byte in the low byte and the most significant byte in the high byte. (This is commonly referred to as "little endian" format.)

A typical response is as follows:

- Record count or sequence number (1 byte)
- Status / error data (1 byte)
- Modbus Message Length (1byte)
- Device's ModBus address (1 byte)
- ModBus command number (1 byte)
- Starting ModBus register address (2 bytes)
- Number of data bytes (1 byte)
- 1st Register data (2 byes)
- 2nd Register data (2 byes)

- 3rd Register data (2 byes)

2.5 How to Write ModBus Device Output Data

1. Configure the port to send output data by setting the size of the poll output buffer at the Master to reflect the number of output bytes that will be sent to the ModBus device (default is 48 bytes).
2. Perform a poll function that has the first byte of the field as the "record number", which the host application must increment each time a new record is to sent to the ModBus device. The next byte will contain the length of the subsequent ModBus data field. The bytes following this will be the ModBus output command string in binary format, to be sent to the ModBus device, less the leading colon, checksum, and carriage return/linefeed.
3. The JDM will check the record number, and if different from the last received record number, will compute the checksum, add the leading colon and convert the command string to ASCII (in ASCII mode), and add the carriage return/linefeed. This message is then sent out the serial port to the addressed ModBus device.
4. If data is to be returned as a result of the command, the JDM will respond with a response with the waiting bit set, and then with waiting bit reset after receiving the ModBus response.
5. When a response is received from the addressed device, the JDM will check the checksum, and if OK, will strip the leading colon and convert the message to binary (in ASCII mode), strip the checksum and carriage return/linefeed.
6. A new record number is generated and placed in the 1st byte of the poll response. The 2nd byte is the receiver error byte and the 3rd byte is the length byte, with the stripped binary reply following. If no error occurs, the error byte will be zero. Otherwise the status byte (2nd byte in the poll response) will have one of the error values shown in Table 2-3.

The data sent to the JDM will **typically** contain the following info:

- Record sequence number (1 byte)
- Status byte (1 byte)
- Length byte (1 byte)
- Device's ModBus address (1 byte)
- ModBus command number (1 byte)
- Beginning Register number (2 byes)
- Number of registers (2 bytes)
- Output data (2 bytes per datum value)

Table 2-3 Poll Response Status Byte Codes

Error Number	Definition
08_{hex}	CRC or LRC error
04_{hex}	ModBus response time-out; a response from the ModBus device did not occur within the time set by the time-out parameter.
02_{hex}	Invalid data received from the ModBus device; data internal to ModBus string is invalid for some reason.

01_{hex}	Invalid ModBus format; for example bad length, no colon in response, etc.
10_{hex}	Invalid ModBus data from DeviceNet master
80_{hex}	Waiting bit; JDM waiting for a response from the ModBus device.

3 Hardware Installation and Configuration

3.1 Overview

The ASCII contains two LEDs to indicate the status of the device and the status of the network. The device can be connected to the main DeviceNet trunk line or to a drop line via screw terminations on the open, pluggable 5-pin DeviceNet connector supplied with your unit. The serial link is connected via an open, pluggable 3-pin connector for RS232 and RS485, or a 5-pin connector for RS422.

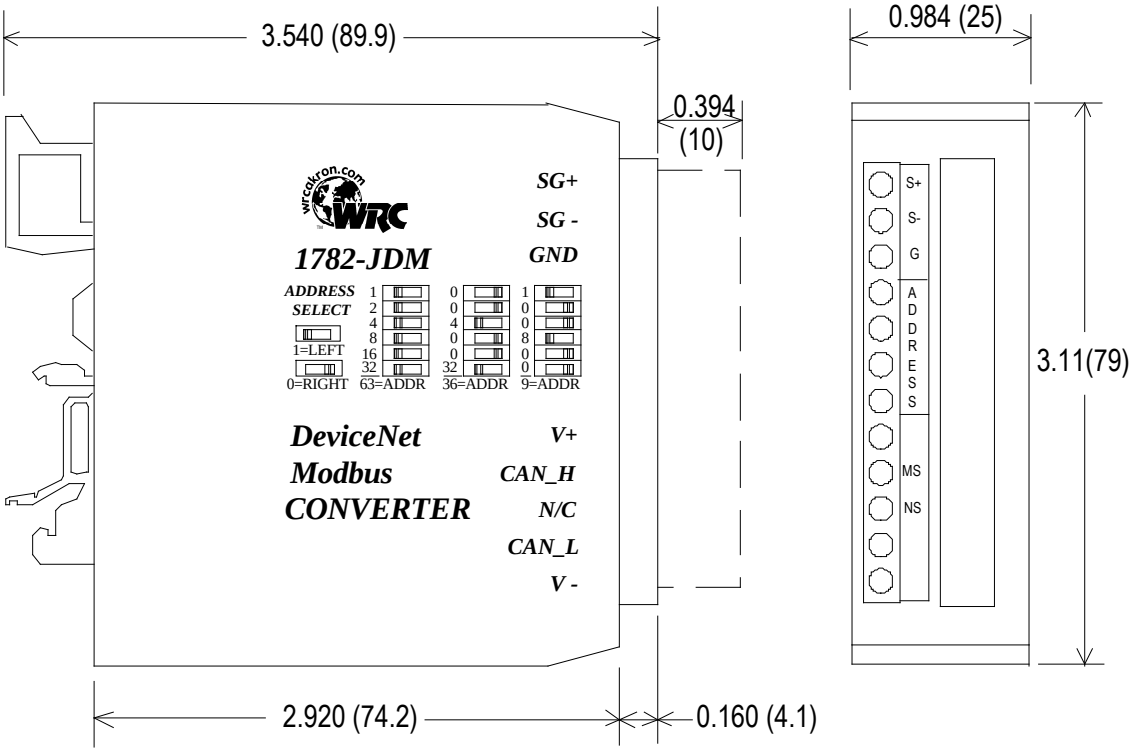

Figure 3-1 1782-JDM Outline

3.2 LED Operation

An ASCII Multiplexer has two LEDs that provide visual status information to the user about the product and the DeviceNet network. See Table 3-1 and Table 3-2.

Table 3-1 Module Status LED (labeled MS)

LED State	Module Status	Meaning
OFF	No Power	There is no power through DeviceNet.
Green	Device Operational	ASCII is operating normally.
Flashing Green	Device in Standby	If in autobaud, ASCII is waiting for a sync signal. (If fixed baud mode, ASCII needs commissioning. Future.)
Flashing Red	Minor Fault	Recoverable fault.
Red	Unrecoverable Fault	ASCII may need replaced.
Flashing Red/Green	Device Self-Testing	ASCII is in self-test mode.

Table 3-2 Network Status LED (labeled NS)

LED State	Module Status	Meaning
OFF	No Power / Not on-line	JDM has no power or has not completed the Dup_MAC_ID test.
Flashing Green	On-line, not connected	JDM is on-line but is not allocated to a Master.
Green	On-line	JDM is operating normally.
Flashing Red	Connection time-out	One or more JDM connections are timed out.
Red	Critical link failure	JDM has detected an error that makes it incapable of communicating on the link. (Bus off or Duplicate MAC ID).

3.3 Serial Port Connectors

Your ModBus device(s) are connected to the JDM via 3-wire or 5-wire communications cable. See your ModBus device User Manual for details on the specific connection.

Table 3-3 RS-232/485 Connector Signals

Pin #	RS232 Signal	RS485 Signal	Designator
1	Receive	Signal -	SIG-
2	Transmit	Signal +	SIG+
3	Ground	Ground	GND

Table 3-4 RS-422 Connector Signals

Pin #	Designator	RS422 Signal
5	TX-	Transmit -
4	TX+	Transmit +
3	GND	Ground
2	RX-	Receive -
1	RX+	Receive +

3.3.1 RS485 Network Termination (1782-JDC-2)

An RS485 system **typically requires a terminator resistor at each end of the line**. Many products may have an optional terminator built into the product. The JDM does not. The typical values for the terminating resistor are:

- 100-120 ohm
- 1% metal film
- 1/4 Watt

This is similar to the termination requirement as for the DeviceNet link (121 ohm), so the same resistors may be used in many applications. It is the user's responsibility to determine the appropriate terminations for the application.

Note: 1782-JDM-2 products shipped with date code "0338" and earlier may have an internal 220-ohm terminating resistor installed. If you have any questions about this, please contact WRC by phone or at support@wrcakron.com.

3.4 Power Requirements

3.4.1 Device power

The JDM is powered directly from the DeviceNet connection 11-25 Vdc power supply. The JDM consumes 90 mA of current at 24 Vdc, or ~2 Watts, typical.

3.5 DeviceNet Network Configuration

DeviceNet specifications provide for a maximum network distances for the main trunk line and drop lines, depending upon the baud rate used on the network. They are:

Table 3-5 Network Maximum Lengths

	Trunk Line Length		**Drop Length**			
	Maximum Distance		**Maximum**		**Cumulative**	
Baud Rate	**Meters**	**Feet**	**Meters**	**Feet**	**Meters**	**Feet**
125k baud	5OO m	164O ft	6 m	2O ft	156 m	512 ft.
25Ok baud	25O m	82O ft	6 m	2O ft	78 m	256 ft.
5OOk baud	1OO m	328 ft	6 m	2O ft	39 m	128 ft.

3.5.1 DeviceNet Network Termination

A DeviceNet system **must be terminated at each end of the trunk line**. The host controller and the **last** JDM or other DeviceNet device on the network must always be terminated to eliminate reflections, even if only two nodes are present. The DeviceNet specifications for the terminating resistor are:

- 121 ohm
- 1% metal film
- 1/4 Watt

An appropriate terminating resistor kit, WRC part number RM121DN, is included with your JDM.

Important: Per the DeviceNet spec -- do not terminate devices on drop lines.

3.5.2 DeviceNet Connection Wiring

The supplied DeviceNet connection plug accepts cable sizes from 12 AWG - 24 AWG. The maximum wire size (12 AWG) has an area of 653O circular mils and the smallest (24 AWG) has an area of 3265 circular mils.

Where not prohibited by local government or wiring regulations or company policy, multiple wires can be inserted each connection point on the plug as long as the total wire area does not exceed that of a 12 AWG wire. Use the chart below as a guide.

Phoenix Contact recommends using the same size wires be used when connecting more than one wire in a screw termination.

UL may require the use of crimped ferrules to connect multiple wires together.

Table 3-6 DeviceNet Wiring Termination

Wire AWG	Wire area (circular mils)	Maximum Wires per Terminal
12	653O	1
14	411O	1
15	3265	2
16	258O	2
18	162O	4 *
2O	1O22	6 *
22	645	9 *
24	4O4	16 *

* WRC does not recommend using more than 2 wires in any wire terminal.

The conductor sizes for DeviceNet cables are:

Table 3-7 DeviceNet Conductor Sizes

Function	Thick Wire	Thin Wire
Power	15 AWG	22 AWG
Signal	18 AWG	24 AWG

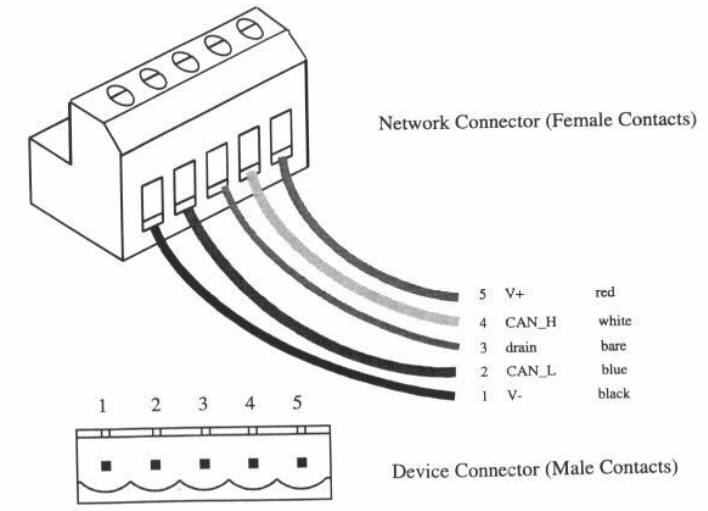

Figure 3-2 DeviceNet cable connector

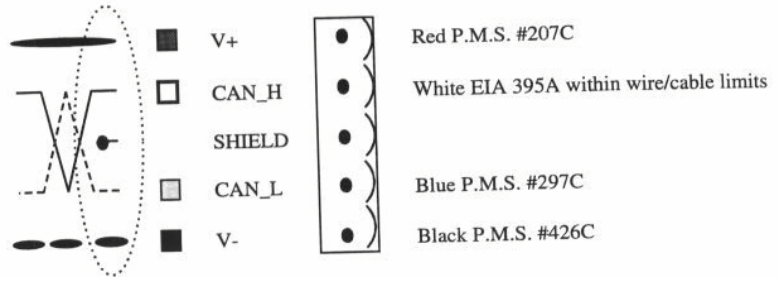

Figure 3-3 DeviceNet cable specifications

4 Software Configuration

The ASCII Multiplexer is software-configured for several parameters. Table 4-1 below defines the legal values and the default values for the device address (referred to as the device's MacID), baud rate and ASCII configuration selections available.

Table 4-1 Configuration Parameters

Parameter	Param. Instance	Description	Parameter Choices		Default Setting	Default Value
ModBus Mode	1	ModBus communication protocol	0 = ASCII, 1 = RTU		ASCII	0
Device Rx Data Size	2	Max size of bytes received from the ModBus device	0 – 100		48	48
Device Tx Data Size	3	Max size of bytes transmitted to the ModBus device	0 – 100		48	48
ModBus Baud rate	4	Serial link communication speed	0 = 9600 1 = 1200 2 = 2400	3 = 4800 4 = 19.2k 5 = 38.4k	9600	0
DeviceNet Baud rate	5	DeviceNet communication speed	0 = 125k, 1 = 250k 2 = 500k, 3 = Autobaud		Autobaud	3
ModBus Time-out Time	6	Timeout in 10 ms ticks for a ModBus response	0 = No time out 1 – 65535 = 1 ms to 65.535 sec		No time-out	00 00
Output Fault Action	7	Action to take upon a fault of the DeviceNet communication link	0 = No Action 1 = Transmit String		No Action	0
Output Fault Value / string	8	The string sent out the port if Output Fault Action is 1.	<user-defined string> (up to 100 bytes)		None	Null
Output Idle Action	9	Action to take upon an Idle condition of the DeviceNet communication link	0 = No Action, 1 = Transmit String		No Action	0
Output Idle Value / string	10	The string sent out the port if Output Idle Action is 1.	<user-defined string> (up to 100 bytes)		None	Null
Serial communication framing format	11	Framing parameters for the serial communication link	0 = 7N2 1 = 7E1 2 = 7O1	3 = 8N1 4 = 8N2 5 = 8E1 6 = 8O1	7N2	0

Definitions of these parameters are as follows:

1. **ModBus Mode**: The JDM can operate in either the ASCII or RTU mode for ModBus communications.

2. **Rx Data Length**: Maximum number of bytes expected from the remote ModBus device. Note that the DeviceNet **produce length** will read this length + 2.

3. **Tx Data Length**: Maximum number of bytes to be sent to the remote ModBus device. Note that the DeviceNet **consume length** will read this length + 1.

4. **ModBus Baud rate**: The communications speed of the RS232 or RS485 link.

5. **DeviceNet Baud rate**: The selection of the baud rate is independent of any other parameter.

6. **ModBus Timeout**: The amount of time the JDM waits for a command response from a ModBus device before it declares an error due to a lack of response from the device. A value of 0 indicates that this feature is not active.

   During while waiting for the ModBus response, if polled, the JDM will return a poll response with the "waiting" bit set.

7. **Output Fault Action**: Selection to determine whether each output will hold its last state or transmit the ModBus string identified in the next parameter upon a device fault.

8. **Output Fault Value/String**: The message string the JDM will output after a Fault, if Fault Value is selected above (hold last state is not selected). The default is no message. This message is sent in "little endian" format.

9. **Output Idle Action**: Selection to determine whether each output will hold its last state or transmit the ModBus string defined in the next parameter if the Master issues an Idle Command.

10. **Output Idle Value/String**: The message string the JDM will output after an Idle Command, if Idle Value is selected above (hold last state is not selected). The default is no message. This message is sent in "little endian" format.

11. **Serial Communications Framing Format**: Selects which format of communications protocol is used, with the valid selection indicated.

Note: "**Little Endian**" format means that the data is formatted such that each 2-byte word of data has the least significant byte in the low order byte of that word and the most significant byte in the high order byte of that word. This is opposite of the actual ModBus string itself. This is done to facilitate the host software program in extracting "word" integer or register data without having to re-order the individual bytes.

5 DeviceNet I/O Assembly Formats

The following tables define the format of the DeviceNet data sent to and received from the JDM. The exact format of the output and input data that is sent from and received at your DeviceNet master is defined by the ModBus command used. The following tables provide the general format and illustrate the differences between the ASCII and RTU modes. Review the ModBus spec or your ModBus device's user's manual for more details.

5.1 Output Data, ModBus Command

There are two types of output command assembly formats. The output command format depends upon whether or not the command data length is less than or equal to 7 bytes or is 8 or more bytes.

If the command length is 8 bytes or more, then a "fragmented message" is created by the host and is sent to the JDM. This will be the more common format for your ModBus command strings and the format that the JDM expects to receive. See Table 5-1.

Table 5-1 I/O Assembly Object Instance 2 Data Format – Fragmented Command

Byte (Offset value)	**Data**
0	Record number
1	Length
2	Address
3	Command
4	Data www – LSB
5	Data www – MSB
6	Data xxx – LSB
7	Data xxx – MSB
…	
N	Last data – LSB
N + 1	Last data – MSB

If the ModBus command string data is 7 bytes or less, then use the command format shown in Table 5-2. This will generate a "non-fragmented DeviceNet message". The 3F (hex) leading byte value indicates to the JDM that this message is a short string that was not fragmented by the sender.

Table 5-2 I/O Assembly Object Instance 2 Data Format – Short Command w/o Fragmentation

Byte (Offset value)	Data
0	Record number
1	Length
2	Address
3	Command
4	Data www – LSB
5	Data www – MSB
6	Data xxx – LSB
7	Data xxx – MSB

5.2 Input Data, ModBus Response

The exact format of the input data that is received at your DeviceNet master is defined by the ModBus command used. The following tables provide the general format and illustrate the differences between the ASCII and RTU modes. Review the ModBus spec or your ModBus device's user's manual for more details.

Table 5-3 I/O Assembly Object Instance 1 Data Format – Response

Byte (Offset value)	Data
0	Record number
1	Status
2	Length
3	Address
4	Function/Command
5	Byte Count
6	Data xxx - LSB
7	Data xxx - MSB
8	Data yyy - LSB
9	Data yyy - MSB
…	
N	Last data - LSB
N + 1	Last data - MSB

For the input assembly to the host, the user does not need be concerned with differentiating between fragmented and non-fragmented messages. The JDM and the host processor will do this transparently.

6 General Specifications

Product: 1782-JDM – DeviceNet-to-ModBus Gateway

- o 1782-JDC-1, Isolated RS-232
- o 1782-JDC-2, Isolated RS-485
- o 1782-JDC-4, Isolated RS-422/485

Description: Communications gateway between a ModBus device over an RS232/422/485 network and a DeviceNet network. The family members include:

- 1782-JDM-1 – Isolated RS-232 serial interface
- 1782-JDM-2 – Isolated RS-485 serial interface
- 1782-JDM-4 – Isolated RS-422/485 serial interface

Device Profile: General Purpose Discrete I/O, Class 7, with objects:

- Identity	(Class 1)
- Message Router	(Class 2)
- DeviceNet	(Class 3)
- Assembly	(Class 4) – 2 instances
- Connection	(Class 5)
- Parameter	(Class F_{hex}) – 9 instances
- ModBus Object	(Class 66_{hex}) – vendor-specific

Communications: Predefined Master/Slave Connection Set, Group 2 Only Server

DeviceNet I/O Protocols: Poll

DeviceNet Baud Rate: Autobaud (default), 125k, 250, 500k - software selectable

DeviceNet Address selection: Address number 1 to 63, switch selectable (default = 63)
Switches 1 – 6
All switches OFF: address = 0
All switches ON: address = 63

DeviceNet Connection: 5-pin pluggable header (male)
Phoenix Contact MSTBA 2.5/5-G-5.08/AU

DeviceNet Cable: 5-contact plug (female contacts) – color-coded
Phoenix Contact MSTB 2.5/5-ST-5.08/AU
(not included, user supplied)

ModBus Connection: 3-pin pluggable header (male)
Phoenix Contact MSTBA 2.5/3-G-5.08/AU

ModBus Serial Cable: 3-contact plug (female contacts) – color-coded
Phoenix Contact MSTB 2.5/3-ST-5.08/AU (not included, user supplied)

DeviceNet: **Baud rate selection:**

Autobaud operation (default)

Fixed baud (software selectable) – 125k, 250k and 500k baud

Address selection:

Address number 0 to 63, switch selectable (default = 63)

Status Indicators: Module Status: green/red bi-color LED

	Network Status: green/red bi-color LED
	Network Termination: External terminating resistor (optional per user system configuration)
Serial port:	**Baud rate**: 1200, 2400, 4800, 9600, 19.2k, 38.4k baud
	software selectable
	Parity: Odd, even or none (software selectable)
	Data bits: 7 or 8 (software selectable)
	Status Indicators: Transmit Active: green LED
	Receive Active: green LED
Voltage Isolation:	1000V network isolation
Maximum power:	200 mA @ 11 Vdc / 90 mA @ 25V dc unregulated power supply
Mounting:	DIN rail mount, EN50022 (WRC50022)
Size:	Depth: 3.54" (90 mm) Width: 0.98" (25 mm) Height: 3.11" (79 mm)
Operating Temp:	0-60 ºC
Humidity:	0-95% RH, non-condensing
Agency Approvals:	UL listed (pending), CE (pending)

7 ModBus Protocol

Complete details about the ModBus protocol can be found in the Modbus Organization web site:

http://www.modbus.org/

8 Troubleshooting

This section identifies some of the common problem observed when commissioning or operating a DeviceNet and JDM.

Problem: JDM powers up.
The module LED flashes green.
The network LED is OFF.
JDM does not communicate over the network.

Possible Solution:
The JDM is in Autobaud mode, and is waiting for a few messages to cross the network so it may determine the baud rate setting. This should happen if any other devices send their Dup Mac messages, or the scanner sends any messages over the network. If the JDM is the only device on the network, it may be advisable to set the baud rate to a fixed speed. Check Parameter 5 in Table 4-1

Problem: JDM powers up.
The network status LED flashes red.
JDM does not communicate over the network.

Possible Solution:
JDM has detected a duplicate MacID (DeviceNet address). Change the JDM's MacID to an unused address and cycle power.

Problem: Module Status LED is solid Green
Network Status LED is flashing Green
Device will not communicate on the network.

Possible Solution:
Network does not have a terminating resistor. Add a 121-ohm resistor across the CAN_H and CAN_L signals at the first and last nodes.

Problem: Module Status LED is solid Green
Network Status LED is solid Green
Device will not perform serial I/O – does not return a poll response and ignores future commands.

Possible Solution:

1. JDM received an unintelligible response from the ModBus device and the time-out is set to 0. Set the time-out value to a non-zero number greater than the longest message time expected from the ModBus device.
2. The JDM is connected to a scanner that only accepts a fixed-length poll response and the ModBus device returned a response of an unexpected length to the JDM. The JDM poll response was rejected by the scanner due to "incorrect size" and took the JDM off-line.

Problem: JDM does not receive a response from the ModBus slave.

Possible Solutions:

1. The ASCII-RTU mode selection is not correct. Check Parameter 1 in Table 4-1.
2. The serial link baud rate does not match the ModBus device baud rate. Check Parameter 4 in Table 4-1.
3. The communication Framing parameters are not correct. Check Parameter 11 in Table 4-1.
4. The ModBus address is not correct in the I/O command.
5. The ModBus device may not support the ModBus specific commands being issued.

Problem: JDM operates on DeviceNet but does not seem to handle ModBus activity.
Possible Solutions:

1. Check the RX and TX LEDs on the JDM. A successful transmission will appear as:
    - Brief TX LED flash
    - Brief RX LED flash
    - Both LEDs off

   If the sequence does not appear as this, then there is a wiring problem.
2. If this is an RS485 link, make sure it is properly terminated. (121ohm resistor at each end.)
3. Make sure the TX and RX, or SIG+ and SIG-, are not crossed.
4. Make sure your signal commons are connected.

9 Accessories and Other WRC Products

The following components can be used with a 1782-JDM-x for replacements or spare parts, or as complementary devices as a part of your DeviceNet system.

Table 9-1 WRC Replacements, Spare Parts and Other Products

Part	WRC Part Number
DIN rail	WRC 50022
Terminating resistor, axial lead	RM121DN
Terminal block connector plug, 18-position	KPELFP18210
Terminal block connector plug, 20-position	KPELFP20210
Discrete I/O block – 4 channels	1782-JDB4
Discrete I/O block – 8 channels	1782-JDB8
Analog Input block – 4 channels, 10-bit	1782-JDA4
Analog I/O block – 8 channels, 12-bit	1782-JDA8
DeviceNet to Serial I/O Gateway	1782-JDC
DeviceNet to ModBus Gateway	1782-JDM
DeviceNet to Pamux Gateway	1782-JDP
DeviceNet to Serial Gateway, open frame	1799wr-DASCII
DeviceNet/CAN Software Utility	DNspector™
Discrete I/O block – 24 channels	W2-JDB24
Discrete I/O block – 48 channels	W2-JDB48
Discrete I/O, Analog Input block – 24 DIO, 32 AI	W2-JDA24
Discrete I/O, Analog Input block – 48 DIO, 32 AI	W2-JDA48
Analog I/O block - 32 channels	W2-JDAIO
Discrete and Analog I/O block – 24 DIO, 32 AIO	W2-JDAIO24
Discrete and Analog I/O block – 48IO, 32 AIO	W2-JDAIO48
Discrete I/O block – 8 DIs, 8 DOs, 4 AIs	W5-JDB16-x
DeviceNet, CANopen Extender, DIN mount	WRC-CANX-DIN-DN
SDS Extender, DIN mount	WRC-CANX-DIN-SD
DeviceNet, CANopen Extender, DIN mount	WRC-CANX-DIN-C7
DeviceNet, CANopen Extender, NEMA box	WRC-CANX-NEM-AU
DeviceNet, CANopen Extender, NEMA box	WRC-CANX-NEM-DN
DeviceNet, CANopen Extender, Fiber Optic, NEMA box, multi-mode fiber	WRC-CANR-DF-DN
DeviceNet, CANopen Extender, Fiber Optic, NEMA box, single-mode fiber	WRC-CANR-DF-SM

10 Frequently Asked Questions

1. ***Where is the 1782-JDMx gateway User's Manual located?***
   The WRC 1782-JDMx product User's Manuals are available from our web site Support -> Technical Data http://wrcakron.com/data.html.

2. ***What are the 1782-JDMx gateways and why should I consider purchasing one for my DeviceNet network?***
   The 1782-JDMx gateways are communication adapter devices that provide you with a flexible DeviceNet-to-ModBus interface for ModBus-capable devices. The JDMx models allow you to easily and conveniently connect and integrate peripheral products with either RS-232, RS-485 or RS-422 serial ports into a DeviceNet system, enabling you to select just the right peripheral device for your application without limiting your selection to DeviceNet-cable devices only.

3. ***How can I change the DeviceNet MAC ID of my 1782-JDMx?***
   The address of the 1782-JDMx will come preset from the factory for address 63. You may change the address using the bank of 6 DIP switches on the unit: they can be set to any address from 0 to 63 in binary (111111=address 63). The MACID or address will not change if the switches are moved while the device is powered; power must be removed and reapplied when a new switch setting and MAC ID are desired or a device reset command must be issued.

4. ***What electrical connections are required for a 1782-JDMx gateway?***
   DeviceNet Cable Connection:
   1, 5-pin pluggable header (male, Phoenix Contact MSTBA 2.5/5-G-5.08/AU or equivalent)

   Serial Cable Connection: For RS-232 & RS-485
   1, 3-pin pluggable header (male, Phoenix Contact MSTBA 2.5/3-G-5.08/AU or equivalent)

   Serial Cable Connection: For RS-422
   1, 5-pin pluggable header (male, Phoenix Contact MSTBA 2.5/5-G-5.08/AU or equivalent)

5. ***How do I begin to use a 1782-JDMx gateway?***
   You can begin to use a 1782-JDMx gateway by following the User's Manual ***Quick Start*** instructions.

6. ***What are the LEDs on the 1782-JDMx indicating?***
   There are two sets of LEDs on the unit. The two green LEDs next to the serial connector are used as indicators for the serial communication receive and transmit ports. The other two bi-color LEDs are used to indicate the DeviceNet module (MS) and network (NS) status. For troubleshooting or further information on status definitions corresponding to LED actions see ***LED Operation*** in the 1782-JDMx User's Manual.

7. ***On what DIN rail can I mount the 1782-JDMx?***
   The 1782-JDMx can be mounted on DIN rail mount, EN 50022 (WRC50022).

8. ***What are some common applications for the 1782-JDMx gateway?***
   A 1782-JDMx gateway provides DeviceNet communications for weigh scales, bar code readers and scanners, display panels, robots, drives, operator stations / HMI, magnetic code readers and other ASCII serial devices.

9. ***Are the 1782-JDMx gateways designed to "plug-n-play"?***
   Yes, the 1782-JDMx gateways are designed to "plug-n-play" using our factory default settings: serial framing of 7N2, serial baud rate of 9600, max number of receive and transmit characters set to 48, Modbus mode set to ASCII, etc. The full list of default settings can be found under ***Default Settings*** in the User's Manual. If you wish to change any parameters from their default settings you will need

to use a DeviceNet software configuration tool. Many of these tools have EDS readers, which make the editing process easier.

10. ***Does a 1782-JDMx gateway provide electrical isolation?***
Yes, a 1782-JDMx gateway provides 1000V network electrical isolation.

11. ***What are the currently available 1782-JDMx gateway models?***
1782-JDM-1: Isolated RS232 Interface
1782-JDM-2: Isolated RS485 Interface
1782-JDM-4: Isolated RS422 Interface

12. ***On what DeviceNet baud rates does the 1782-JDMx communicate?***
The unit works with 125k, 250k or 500k baudrates. It is default set to autobaud on DeviceNet. A fixed baudrate of 125k, 250k or 500k may also be assigned with a software configuration tool.

13. ***On what baud rates does the serial port communicate?***
The serial port on the 1782-JDMx will communicate on baudrates of: 1200, 2400, 4800, 9600, 19.2k, or 38.4k baud.

14. ***How do I connect a 1782-JDMx gateway to a serial port device?***
You connect the JDMx to a serial port Modbus device with the Phoenix connectors on the unit, which are labeled for which signal goes in each port of the connector. For RS-232 or RS-485 communications the signal/pin assignments are as follows:

Pin #	RS232 Signal	RS485 Signal	Designator
1	Receive	Signal -	SIG-
2	Transmit	Signal +	SIG+
3	Ground	Ground	GND

For RS-422 communications the signal/pin assignments are as follows:

Pin #	Designator	RS422 Signal
5	TX-	Transmit -
4	TX+	Transmit +
3	GND	Ground
2	RX-	Receive -
1	RX+	Receive +

15. ***How is the Master Handshake mode different from the Hardware handshake RTS/CTS?***
Master Handshake mode is a data control mechanism between the DeviceNet master and the 1782-JDMx (after the JDMx receives data from serial device), communicated in IO message. The 1782-JDMx indicates new data arrival and waits for Master to issue a new record number, to be used with new data. The hardware handshaking via RTS/CTS (which is not supported in the 1782-JDMx) requires two additional signal lines on the serial side and it controls the data before it is received.

16. ***When should I turn Pad Mode ON? Or what configuration parameter should I check when the scanner cannot stay connected to the 1782-JDMx due to an IO size error?***
If the DeviceNet scanner or the master device needs to be pre-configured with an input (produce) size before making a connection, or if the serial device will be sending variable data size strings, you

will need to turn the PAD Mode ON. This parameter should be checked if the 1782-JDMx will not stay connected due to an IO size error.

17. ***What should my master device logic be doing to see if new data is present in a poll message?***
The Master program logic should be looking at the Receive Record Number and comparing it with the last stored record number to monitor new data arrival in Immediate (Non-Handshake) Mode. In the Master Handshake mode, the program should be looking at the New Data bit in status and if detected, should supply the new record number and wait for the 1782-JDMx to produce data along with that record number.

18. ***What does the 1782-JDMx gateway require for power?***
The 1782-JDMx receives all necessary power from the DeviceNet network cable. This should be between 11 and 25 Vdc. It consumes about 2 watts of power maximum.

19. ***How do I access the serial ASCII data from a 1782-JDMx gateway?***
When configuring your PLC, you have the option to memory map the ASCII data from the 1782-JDMx to a location on your PC. The ASCII data can be found at this location.

20. ***When do I need to use an EDS (electronic data sheet) file for a 1782-JDMx gateway?***
An EDS file is not *necessary*, but it makes configuring the 1782-JDMx (changing baud rate, framing, etc.) far easier than if you were to try to change the device parameters by editing the attributes of certain class objects and instances without the aid of a configuration tool. There are many software configuration tools available with EDS readers that can be used to set up the device.

21. ***What is the proper cable to use for a 1782-JDMx gateway serial connection?***
The proper cable selection depends upon which gateway model is used, the serial communication baud rate desired and the total cable length. Cable selection guidelines:

1782-JDM-1: RS232 – 3-wire shielded
1782-JDC-2: RS485 – 3-wire shielded
1782-JDC-4: RS485/RS422 – 3-wire (RS485) shielded or 5-wire (RS422) shielded

22. ***Where can I purchase the proper cable for a 1782-JDMx gateway serial connection?***
You can purchase the proper cable from many suppliers and fitted with the connectors required for the gateway model and serial device being used.

If the answer to your question is not found here, or you require further assistance, please contact Western Reserve Controls at:
1485 Exeter Road
Akron, OH 44306
330-733-6662 (phone)
330-733-6663 (fax)
For product information, e-mail: sales@wrcakron.com
For technical support, e-mail: support@wrcakron.com